AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

1. (Previously Presented) A method of effecting secure communications between a server and a client, the server executed in a server computer, the method comprising:

detecting, at the server computer, a client connection at a first port;

providing, by the server computer, the client with a decoy port number; and

providing, by the server computer, services to the client on a second port having a second port number that is mapped to the decoy port number, wherein the second port number is different from the decoy port number.

2. (Previously Presented) A method as defined in Claim 1, wherein the decoy port number is provided to the client by the operation of a routine that is associated with the server, the routine executed in the server computer.

3. (Original) A method as defined in Claim 2, further comprising:

launching the server on the second port; and

monitoring the second port for a connection by the client.

4. (Original) A method as defined in Claim 3, further comprising;

if there is no connection by the client within a predetermined time interval, terminating execution of the server on the second port.

5. (Previously Presented) A method as defined in Claim 2, further comprising:

maintaining, in the server computer, a table of available decoy port numbers that are mapped to valid port numbers.

6. (Previously Presented) A method as defined in Claim 5, wherein the table maintained in the server computer corresponds to a second table maintained at a client computer on which the client is executed, the second table mapping decoy numbers to valid port numbers at the client computer.

7. (Original) A method as defined in Claim 6, further comprising:
monitoring the second port for a connection by the client, and
if there is no connection by the client within a predetermined time interval, terminating execution of the server on the second port.

8. (Cancelled)

9. (Previously Presented) A computer system comprising:
a plurality of ports, each port having a respective port number;
a server application; and
a routine that, if executed, is operative to:
detect a client connection at a first port;
provide the client with a decoy port number; and
provide services to the client on a second port having a second port number that is mapped to the decoy port number, wherein the second port number is different from the decoy port number.

10. (Original) A computer system as defined in Claim 9, wherein the routine, if executed, is operative to:
launch the server application on the second port; and
monitor the second port for a connection by the client.

11. (Original) A computer system as defined in Claim 10, wherein the routine, if executed, is operative to terminate execution of the server application on the second port if there is no connection by the client within a predetermined time interval.

12. (Previously Presented) A computer system as defined in Claim 9, wherein the routine, if executed, is operative to maintain a table of decoy port numbers mapped to corresponding valid port numbers.

13. (Original) A computer system as defined in Claim 12, wherein the routine, if executed, is operative to:

launch the server application on the second port subsequent to providing the decoy port number to the client.

14. (Cancelled)

15. (Previously Presented) A server computer system comprising:

a plurality of ports, each port having a respective port number;

a first server application; and

a first routine that is associated with the first server application and that, if executed, is operative to:

detect a client connection at a first port;

transmit a decoy port number to the client;

terminate the connection to the first port; and

provide services to the client on a second port having a second port number that is mapped to the decoy port number, the second port number being a valid port number that is different from the decoy port number;

a second server application; and

a second routine that is associated with the second server application and that, if executed, is operative to:

detect a client connection at a third port;

transmit a second decoy port number to the client;

terminate the connection to the third port; and

provide services to the client on a fourth port having a fourth port number that is mapped to the second decoy port number, the fourth part number being

another valid port number that is different from the second decoy port number.

16. (Previously Presented) A server computer system as defined in Claim 15, wherein the first routine and the second routine, if executed are operable, respectively, to:

terminate execution of the first server application on the second port if there is no client connection within a predetermined time interval; and

terminate execution of the second server application on the fourth port if there is no client connection within a predetermined time interval.

17. (Previously Presented) A method executed by a client computer, comprising:

attempting to access a server application on a first port of a server computer;

receiving, from the server computer, a decoy port number that is an invalid port number;

translating the decoy port number to a valid port number; and

connecting to the server application on the valid port number.

18. (Previously Presented) A method as defined in Claim 17, wherein the decoy port number is translated using a wrapper script associated with a client application in the client computer.

19. (Previously Presented) A method as defined in Claim 17, wherein the decoy port number is translated using code embedded in a client application in the client computer.

20. (Previously Presented) A method as defined in Claim 17, further comprising:

mapping the decoy port number to an intermediate port number; and

effecting an offset to the intermediate port number to produce the valid port number.

21. (Previously Presented) A computer system comprising:

a plurality of ports, each port having a respective port number;

an application; and

means for effecting secure access to the application by redirecting a client from a first port to a second port, wherein the means for effecting secure access comprises:

a routine that, if executed, is operable to provide the client with a decoy port number that maps to a second port number of the second port, wherein the decoy port number is an invalid port number and the second port number is a valid port number.

22. (Cancelled)

23. (Previously Presented) An article comprising a machine-readable storage medium that comprises instructions that, if executed, cause a server computer to:

detect a connection at a first port of the server computer by a client application;

transmit, to the client application, a decoy port number, wherein the decoy port number is an invalid port number; and

cause a server application in the server computer to be launched at a second port that has a second port number mapped to the decoy port number, the second port number being a valid port number.

24. (Original) An article as defined in Claim 23, further comprising instructions, that, if executed, are operable to:

monitor the second port; and

if there is no connection by the client application within a predetermined time interval, terminate execution of the server on the second port.

25. (Original) An article as defined in Claim 23, wherein the storage medium further comprises a table of decoy port numbers that are mapped to valid port numbers.

26. (Cancelled)

27. (Previously Presented) A client/server system comprising:

a server computer system; and

a server application installed on the sever computer system and comprising instructions that, if executed on the server computer system, are effective to:

detect a connection at a first port by a client application;

transmit, to the client application, a decoy port number, wherein the decoy port number is an invalid port number;

terminate the connection on the first port; and

provide services to the client application on a second port having a second port number that is mapped to the decoy port number.

28. (Previously Presented) A client/server system as defined in Claim 27, further comprising:

a client computer system; and

a client application installed on the client computer system and comprising instructions that, if executed on the client computer system, are effective to:

attempt to access the server application on the first port;

translate the decoy port number to the second port number; and

connect to the server application on the second port.

29. (Cancelled)

30. (Previously Presented) A client/server system as defined in Claim 28, wherein the client application further comprises instructions that, if executed on the client computer system, are effective to:

map the decoy port number to an intermediate port number; and

impart an offset to the intermediate port number so as to derive the second port number.

31. (Previously Presented) The method as defined in Claim 1, wherein providing the decoy port number comprises providing the decoy port number that has no meaning to an

unauthorized client computer, but the decoy port number is mappable to the second port number by an authorized client computer.

32. (Previously Presented) The computer system as defined in Claim 12, wherein the decoy port number provided to the client enables the client to map, using a second table associated with the client, the decoy port number to the second port number such that the client can connect to the computer system at the second port number.

33. (Previously Presented) The computer system as defined in Claim 9, wherein the decoy port number has no meaning to an unauthorized client computer, but the decoy port number is mappable to the second port number by an authorized client computer.

34. (Previously Presented) The article of Claim 23, wherein the decoy port number is meaningless to an unauthorized client computer, but the decoy port number is mappable to the valid port number by an authorized client computer.